

Handling temporal constraints in interaction protocols for intelligent multi-agent systems

Awais Qasim^{1,2,*}, Sobia Iqbal¹,
Zeeshan Aziz², Syed Asad Raza
Kazmi¹, Adeel Munawar³,
Basit Ali Gilani³ and Neelam Qasim⁴

¹Department of Computer Science,
GC University, Lahore, Pakistan.

²School of Science, Engineering
and Environment, University of
Salford, Salford, UK.

³Department of Computer Science,
Lahore Garrison University, Lahore,
Pakistan.

⁴Lahore Business School,
University of Lahore, Lahore,
Pakistan.

*E-mail: Awais@gcu.edu.pk

This paper was edited by
Subhas Chandra Mukhopadhyay,
Anindya Nag and
Nasrin Afsarimanesh.

Received for publication
February 21, 2020.

Abstract

This research focuses on handling temporal constraints in interaction protocols for multi-agent systems. There is a dire need of standardized interaction protocols that can be used to handle timing aspects in real-time multi-agent system's negotiation. The most commonly used Foundation for Intelligent Physical Agents Protocol lacks the appropriate specification in this regard. In real-time systems timing constraint is a major concern for all of its tasks and goals. Agents require real-time responses and must eliminate the possibility of massive communication between them. The timing specification of these real-time multi-agent systems in which agents communicate with each other to achieve their goals within deadline will be of great value for their correct functioning. A high degree of dependability and predictability is expected from real-time software agents. The basis of our work is the standardized interaction protocols to support the communication between agents in real-time environment and this is possible via message passing. By incorporation of well-defined timing parameters in Foundation for Intelligent Physical Agents performatives, we have enabled them to be used in any real-time multi-agent's communication. We demonstrate the usage and effectiveness of our proposed real-time performatives using a case study of monitoring boats in marine reserves in which the agents interact with each other to accomplish their goals.

Keywords

Multi-agent systems, Real-time systems, FIPA performatives, Agents communication, Real-time communication, Interaction protocols.

There is a dire need of standardized interaction protocols that can be used to handle timing aspects in real-time multi-agent systems (RTMAS) negotiation. In real-time systems, time constraint is the major responsibility for all of its tasks and goals. Multi-agent systems (MAS) with collaborative agents must be able to interact and communicate with each other. Agents modeling is appropriate for message passing between interacting agents to achieve some goals. For the cooperation and coordination in multi-agent system, multiple agents operate in the environment. Communication between multiple agents leads with the set of rules that is called interaction protocols.

These interaction protocols are guaranteed for interoperability between agents by coordination. Agent accomplishes its task in a particular domain. In short, RTMAS utilize this approach which is suitable for solving complex problems that require response time. Julian et al. (2002) proposed a SIMBA (Sistema Multiagente Basado en ARTIS) architecture that was based on agent technology for RTMAS. It is based on ARTIS agents with critical temporal constraints (Botti et al., 1999). This architecture specifies a multi-agent platform for real-time agents that perform time bounded actions. SIMBA systems are FIPA (Foundation for Intelligent Physical Agents) compliant

for distributed domain and allow communication between agents with hard temporal constraints. Stankovic (1988) presented a real-time system in which accurate functioning not only depends on the computational results but also depends on the time at which the results are produced. Real time agent (RTA) defines itself with restrictions of timing constraints on its responsibilities. These restrictions may or may not be critical. RTAs collaboratively work in real-time environment to achieve common goals with less communication for timely response. An MAS with single RTA is called an RTMAS. The performance of RTA depends on task completion within deadline. In the study of Julian and Botti (2004), classification of RTAs is defined as hard real-time agents and soft real-time agents. Soft real-time agents deal with minor relaxed threshold for achieving their temporal restrictions and hard real-time agents enforce that task should be completed within deadline. MAS with collaborative agents achieve the common goal through negotiation. The most commonly used FIPA Protocol lacks the appropriate specification of calculating the duration of completed tasks and it forms the basis of our work. We enhance the standardized interaction protocols to support the communication between agents in real-time environment enabling them to calculate the duration of the completed tasks using parameterized message passing. The proposed approach is suitable for a large number of application domains including but not limited to real-time traffic monitoring, disaster management systems, surveillance systems, smart grids, etc. By incorporation of well-defined timing parameters in FIPA performatives, we have enabled them to be used in any real-time multi-agent's communication. Interaction protocols with its complexity take long time period for communication in real-time domain. These complex interaction protocols are not suitable for the problems with time bounded tasks. By using RTMAS, message flow between multiple RTAs is controlled by interaction protocols. These interaction protocols define the sequence of messages, number of messages, and updation. FIPA specifications give the outline for architecture and actions for agents. Agent communication language (ACL) is a proposed standard language by FIPA for the interaction between agents like FIPA-ACL. This language has a defined set of performatives that are used when agents do message passing during communication. In this research, existing FIPA performatives are modeled as real-time FIPA performatives. In the specification of RTMAS, agents interact with one another for their goals within timing specification. This element is required for their correct functioning. Protocols define

how much long period of time the agents would wait for the concerned interacting agents.

The rest of the paper is organized as follows. In the second section, we briefly explain the related work relevant to this research. The third section describes the proposed real-time FIPA performatives. The fourth section presents the demonstration of the proposed real-time FIPA performatives using a case study of scheduling analysis and correction for non-schedulable tasks. The fifth section concludes the paper.

Related work

According to Bhouri et al. (2012), a bimodal urban traffic control strategy based on a multi-agent model in which traffic regulation is obtained from the heterogeneous agents that represent the urban network communicate among themselves. They collaborate and negotiate to solve traffic regulating problems and bus regularity compared to a fixed-time control strategy. However, their work did handle the real-time constraints. Hosny Abbas and Shaheen (2015) proposed a multi-layered agent-based industrial control networks (ICN) in which agents cooperate to provide an effective supervision and control of a set of control processes, controlled by a set of legacy control systems with limited computing capabilities. Linnenberg et al. (2011) presented a decentralized market-based power control system with implementation. This gives control of generators and consumers of electrical energy for multi-agent systems that are market based and decentralized. Both of these works focused on agent's cooperation but not at the interaction protocol level. Cintuglu et al. (2015) specified a multi agent framework in which a laboratory-based smart grid test bed is implemented by using IEC 61850 and FIPA standards. This implemented framework can share common information between two platforms by adoption of open connectivity unified architecture OPC interface. In the study of Kristensen and Smith (2015), this describes an agent technology-based traffic system that is simulated by JADE. Autonomous agents work together for traffic simulation in an intersection by FIPA official standards and represent cars and traffic lights working in the traffic system. This presents automated negotiation based on self-interested agents that defines as MAS. It can give a flexible price instead of a fixed price in e-commerce can maximize the payoffs of both buyer and seller (Kexing, 2011). In all of these approaches, there is no mechanism to calculate total duration of the completed tasks. The main purpose of Reid and Shakshuki's (2017) study present the architecture and design of prototype

implementation of an MAS to facilitate healthcare for elderly citizens. Prototype application presents to monitor and advertise health information about patients to healthcare professionals. Li et al. (2016) presented a Smart Home concept, that is MAS-based decentralized architecture allowing residents to keep high-level comfort with effective use of electricity and is aimed to optimize efficiency and cost savings of the homes. It is a demand side management strategy which was integrated into the existing home energy management system (HEMS). In the study of Jemal et al. (2015), a medical multi-agent system (MMAS) that handles problem which appear in the hospital for collaboration between hospital wards, explanation of medical diagnostics, coordination among medical entities and the collection of information about patients. It provides better healthcare than the traditional medical system. Bhardwaj et al. (2014) specified agent-based distributed railway system, that is focused on the issue of passing the trains in same direction by taking decisions through train agents negotiation and reduce the system delay for acceptable limit. In the study of Shpilevoy et al. (2013), agent-based method of adaptive resource scheduling that presents functionality of multi-agent system-based smart factory. It also manages real-time resources in manufacturing workshops. The real-time resource management in manufacturing of aircraft jet engines is presented. According to Elshaafi et al. (2018), MAS that can be connected to other components in smart grid for manage flexibility within the low voltage part of the electricity in network. This approach is used where decision making is decentralized. Agent-based home energy management that provides automated demand response. Ghosn et al. (2010) has implemented MAS in JADE, providing help in smart grid problems and in the improvement of electrical grid in different ways by simulation. All of these approaches are for a specific domain and no generic solution has been discussed. Across the distributed environments, autonomous agents can communicate with each other. Cao (2012) specified the research based on automated negotiation. The agent's negotiation architecture model describes a process that supports the negotiation of aircraft purchase and demonstrates effectiveness of negotiates in an e-commerce environment. Their approach is not suitable for real-time tasks as the decision-making process in ecommerce can take a lot of time. In the study of Filgueiras et al. (2012), an execution model that makes possible of task completion of a mobile agent within deadlines in distributed systems, spatially in real-time scenarios has been discussed.

It provides a mechanism for real-time scheduling in JADE platform. Ghorbani et al. (2012) specified the working of decentralized MAS with distributed power for detection of error in real-time environments. Their work focuses on task scheduling and error detection for MASs in distributed environment but not at the interaction protocol level. The work of Qasim et al. (2015, 2019, 2020) and Qasim and Kazmi (2016) leads to formal specification and verification of interactive real-time software agents (RT agents). Agents work independently and handle the uncertain scenarios. Visually expressive broader structure and modeling approach, i.e. TAPN have been used for specification and representation of stock market system (SMS). It is based on RTMAS. The model is verified by Timed Computational Tree Logic (TCTL) fragments AF, AG, EG, and EF. In this paper, KQML register conversation and simple negotiation interaction conversation are modeled through CPN.

Proposed real-time FIPA performatives

Request performative

In Figure 1, the standardized FIPA request interaction protocol is presented. The sender requests to the recipient for executing some action. Sender initiates

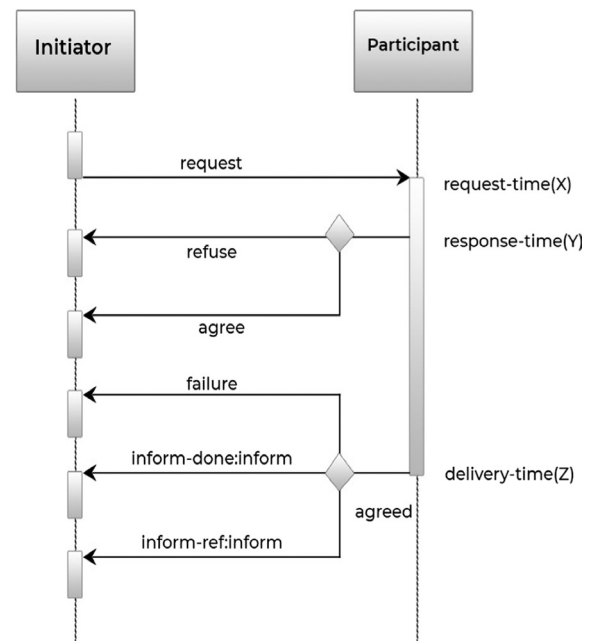


Figure 1: FIPA request interaction protocol.

a request within request-time (X). As a response to this request, a recipient may agree to or refuse the request within a specific time that is called response-time (Y). If the request has been agreed, then recipient will respond within delivery-time (Z) and this delivery-time will be utilized by the messages with inform or inform-ref or failure performatives. A request message should be delivered within specific time that is called request completion time in which a message is delivered to the recipient. Message should be delivered within specific time that is called completion-time.

The total time for the message will be $X + Y + Z$.

```
(request
:sender A
:receiver B
:request-time(X)
:response-time(Y)
:delivery-time(Z)
:protocol FIPA-request
:language FIPA-sl)
```

Agree/refuse/not-understood performative

A message with agree/refuse/not-understood performative indicates the response-time (Y) in which recipient agree/refuse/not-understood to perform given action with any reason.

```
(agree/refuse/not-understood
:sender A
:receiver B
:response-time(Y)
:protocol FIPA-request
:language FIPA-sl)
```

Failure/inform/inform-if/inform-ref performative

A failure message indicates the delivery-time (Z) in which sender inform another agent that action is failed. A message with inform/inform-if/inform-ref performative indicates the delivery-time (Z) in which sender informs the recipient that the content is true or content may be true or false or content with value.

```
(failure/inform/inform-if/ inform-ref
:sender A
:receiver B
:delivery-time(Z)
:protocol FIPA-request
:language FIPA-sl)
```

Query-IF/query-ref performative

In Figure 2, the standardized FIPA query interaction protocol is presented. The action in which sender asks recipient about certain conditions or contents that these are true or false. In message with query-if/ query-ref performative, a sender asks from recipient about certain proposition within specific time that is called query-time (X). As a response, this proposition will be true or false within response-time (Y). If proposition is true, then recipient agrees to complete the request within delivery-time (Z) and this time will be utilized by the messages with inform-if or inform-ref or failure performatives. Message should be delivered within specific time that is called completion-time. The total time for the message will be $X + Y + Z$.

```
(query-if/query-ref
:sender (A)
:receiver (B)
:query-time(X)
:response-time(Y))
:delivery-time(Z)
:language FIPA-sl)
```

Request-when performative

In Figure 3, the standardized FIPA request when interaction protocol is presented. One agent requests

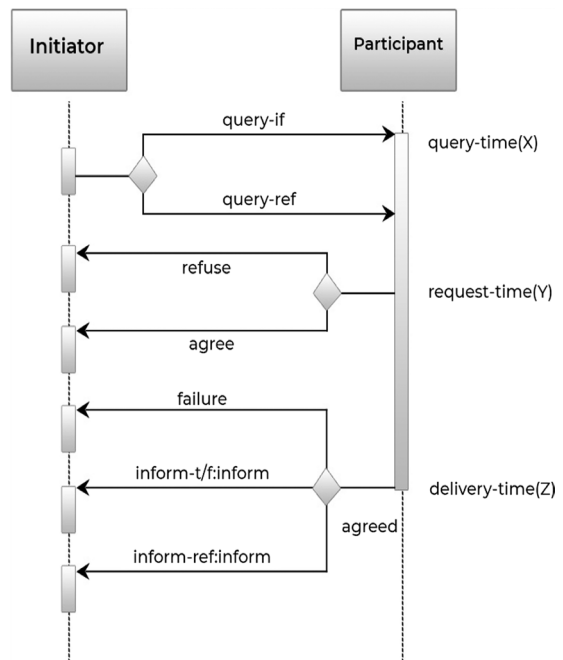


Figure 2: FIPA query interaction protocol.

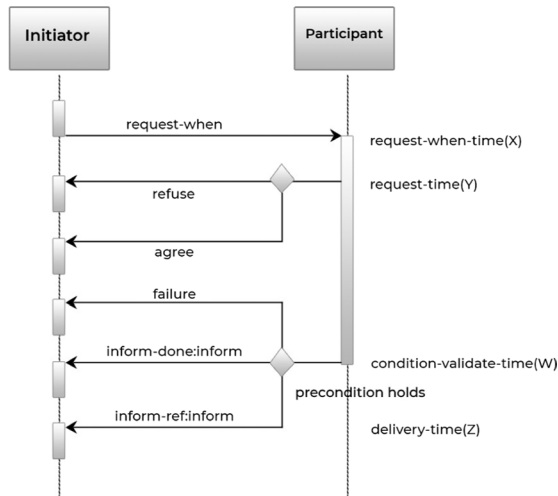


Figure 3: FIPA request when interaction protocol.

the other agent to make an action when given proposition becomes true and in case of refuse further execution will not be proceeded. The agent receives a request from another agent within request-when-time (X) and it can be either refuse or agree within response-time (Y). If recipient agrees then it makes sure that the action is performed when precondition is true. Then recipient will wait within condition-validate-time (W). As soon as precondition holds, recipient will respond within delivery-time (Z) and this time will be utilized by the messages with inform, inform-ref and failure performatives. Message should be delivered with in specific time that is called completion-time and it will be $X + Y + W + Z$.

```

(request-when
:sender (A)
:receiver (B)
:request-when-time(X)
:response-time(Y))
:condition-validate-time(W)
:delivery-time(Z)
:language FIPA-sl)
  
```

Request-whenever performative

In request, whenever a sender requests a receiver to make an action whenever the proposition that is expressed as precondition is true. The agent receives a request from an agent within request-whenever-time (X) and it can be either refuse or agree within response-time (Y). If recipient do not agree to perform action when precondition becomes false then recipient

reevaluates the precondition within re-evaluate-time (V), takes action when its value changes. If the recipient agrees to perform action when the precondition becomes true, then it will wait within condition-validate-time (W). As soon as precondition holds and recipient will respond within delivery-time (Z) and time will be utilized by the messages with inform, inform-ref and failure performatives. Message should be delivered with in specific time that is called completion-time and it will be $X + Y + W + V + Z$.

```

(request-whenever
:sender (A)
:receiver (B)
:request-whenever-time(X)
:response-time(Y))
:condition-validate-time(W)
:re-evaluate-time(V)
:delivery-time(Z)
:language FIPA-sl)
  
```

Cancel performative

At any time during the interaction, the initiating agent may cancel the interaction within cancel-time (C) by sending a cancel message.

```

(cancel
:sender (A)
:receiver (B)
:cancel-time(C)
:language FIPA-sl)
  
```

CFP performative

In Figure 4, the standardized FIPA contract net protocol is presented. A call for proposal (CFP) performative is used to perform a given action. Manager sends CFPs to contractors within call-time (X). Contractors send the reply in three forms that is not understood or refused or proposal within response-time (Y). After receiving the response of contractors, at the same time manger approves one of the contractor's proposal within accept/reject-proposal-time (p) and also sends the rejection of proposal within accept/reject-proposal-time (p) to remaining agents. The contractor sends reply to manager after completion within delivery-time (Z) and this time will be utilized by the messages with inform or inform-ref or failure performatives. Agent issues a call for proposal act that specifies a task within CFP-time that is $X + Y + P + Z$.

```

(cfp
:sender (B)
:receiver (A)
  
```

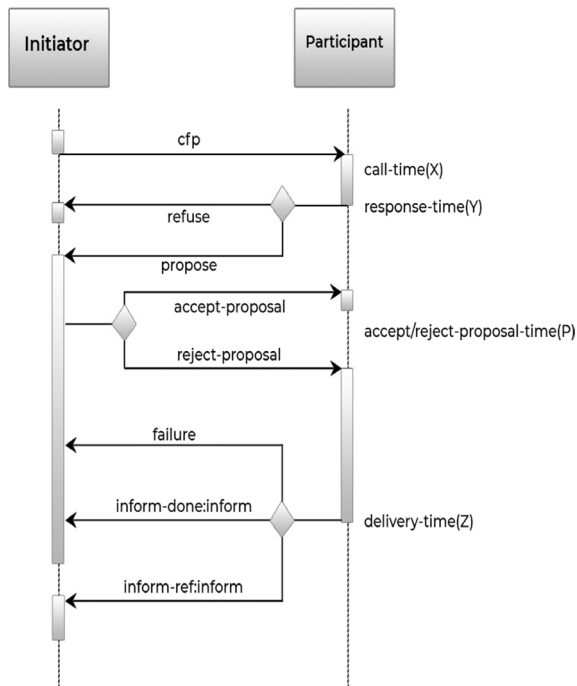



Figure 4: FIPA contract net interaction protocol.

:call-time(X)
:response-time(Y)
:accept/reject-proposal-time(P)
:delivery-time(Z)
:language _pa-sl)

Propose performative

In Figure 5, the standardized FIPA propose interaction protocol is presented. It is used as a response to a CFP. The action of submitting a proposal within response-time (Y) to perform an action by given

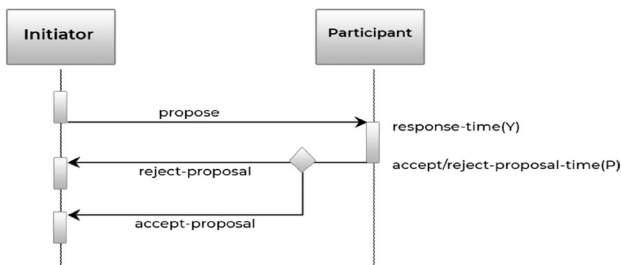


Figure 5: FIPA propose interaction protocol.

preconditions. Some recipient may refuse to propose within response-time (Y).

```
(propose
:sender (B)
:receiver (A)
:response-time(Y)
:language fipa-sl)
```

Accept/reject proposal performative

An agent will accept or reject the proposal of another agent that is submitted previously to perform an action within accept/reject-proposal-time (P). When deadline is finished after receiving the CFP then sender agent checks the received proposals and sends messages to the selected agents for accept proposal within accept/reject-proposal-time (P). At the same time, it sends messages to the remaining agents for reject proposal within accept/reject-proposal-time (P). Recipient do not want to performs the action within accept/reject-proposal-time (P) under the given preconditions.

```
(accept/reject-proposal
:sender (A)
:receiver (B)
:accept/reject-proposal-time(P)
:language FIPA-sl)
```

Subscribe performative

In Figure 6, the standardized FIPA subscribe interaction protocol is presented This allows for requesting

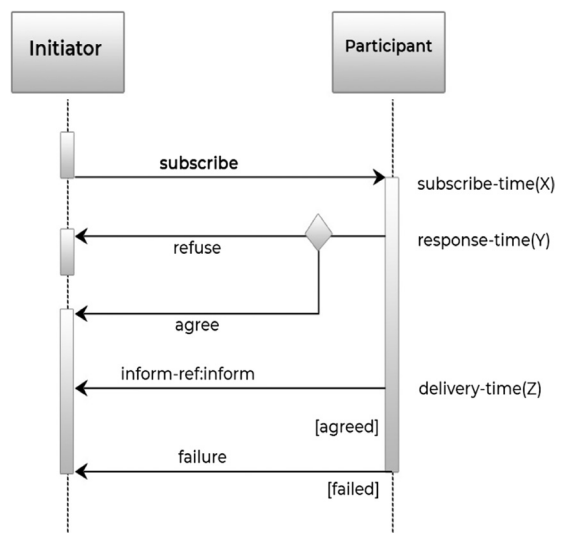


Figure 6: FIPA subscribe interaction protocol.

agents to notify the sender about subscription within subscribe-time (X). After that it is decided whether to accept or refuse the query request within response-time (Y). After the refuse decision of recipient, communication is dismissed. If recipient agrees to perform the action, then its responded within delivery-time (Z) and this time will be utilized by the messages with inform-ref performative. There is a point after the recipient agrees, when it becomes fail to perform action then it sends a message with failure action within delivery-time (Z) that also terminates the interaction.

```
(Subscribe
:sender (A)
:receiver (B)
:subscribe-time(X)
:response-time(Y)
:delivery-time(Z)
:language FIPA-sl)
```

Confirm performative

The sender informs the receiver within confirmation-time (F) about proposition which is true, whereas receiver has the uncertainty about the proposition.

```
(confirm
:sender (A)
:receiver (B)
:confirmation-time(F)
:language fipa-sl)
```

Disconfirm performative

Sender agent informs the receiver agent about the false state of proposition within dis-confirmation-time (G) but in this case receiver has believed the true state of proposition.

```
(disconfirm
:sender (A)
:receiver (B)
:disconfirmation-time(G)
:language FIPA-sl)
```

Demonstration of the proposed real-time FIPA performatives

In this section, we demonstrate the usage and effectiveness of our proposed real-time performatives using two real-time case studies in which the agents interact with each other to accomplish their goals.

Case study of scheduling analysis and correction for non-schedulable tasks of real-time multi-agent systems

Most practicing thing for multiprocessor real-time systems is partitioned multiprocessor scheduling. In the study of Mahfoudhi et al. (2018), the scheduling analysis and correction is used to correct the non-schedulable partition for reducing the complexity. There are two types of agents. First one is initiator as allocator agent (AA) that is a manager and capable to manage the communication between the processor agents. Second one is participant as processor agent (PA) that specifies one processor which is responsible to analyze the schedule ability of its partition. So there are four processors that include processor agent 1 (PA1), processor agent 2 (PA2), processor agent 3 (PA3), and processor agent 4 (PA4). On the basis of priority scheduling analysis, these processor agents are ready to analyze its partition. If the analysis of partition fails and this partition is non-schedulable then correction process begins.

Scenario (correction for non-schedulable partitions P2 and P4)

There are four partitions P1, P2, P3, and P4, and one agent is associated with each partition. These are PA1, PA2, PA3, and PA4. Two partitions P1 and P3 are indicated as schedulable. P2 and P4 are declared as non-schedulable partitions after receiving failed analysis of its partition. PA2 calls the AA to reallocate the task T14 and PA4 calls the AA to reallocate the task T12. The agent's interaction and created contracts and deals are described by correction process that will be triggered. It includes 15 independent tasks that are running on four same processors.

Task = T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T14, T15

Processes = P1, P2, P3, P4

The tasks allocations by processors are described as follows.

Tasks Allocation

T1	P1
T2	P2
T3	P1
T4	P3
T5	P4
T6	P2
T7	P1
T8	P4
T9	P1

T10 P2
T11 P4
T12 P4
T13 P3
T14 P2
T15 P3

The four processors are working with different capacities:

$U1 = 0.8125$
 $U2 = 1.1111$
 $U3 = 0.8000$
 $U4 = 1.1428$

The task allocation for:

Partition P1=(T1, T3, T7, T9)
Partition P2=(T2, T6, T10, T14)
Partition P3=(T4, T13, T15)
Partition P4=(T5, T8, T11, T12)

Partitions with failed analysis:

Partition P2=(T2, T6, T10, T14)
Partition P4=(T5, T8, T11, T12)

The task T14 and task T12 are declared as non-schedulable after failed analysis of its partition P2 and P4. PA2 calls the AA to reallocate the task T14 and PA4 calls the AA to reallocate the task T12. First, AA selects a task T12 because its utilization factor

is highest than utilization factor of T14, and second, AA selects the task T12 for rescheduling. This process will be iterated until the re-allocation of all the tasks.

T12 ($u_{12}=0.1428$)
T14 ($u_{14}=0.1111$)

As shown in Figure 10, PA4 calls the AA to reallocate the task T12 and sends it as a CFP to the PA1 and PA3. T12 ($u_{12}=0.1428$)

PA1 calculates the new capacity of processor1 by including task T12. If capacity is not exceeded, then PA1 responses as propose to schedule T12

New $U1 = [U1 + U12] = [0.8125 + 0.1428] = 0.9553$.

PA3 also calculates the new capacity of processor3 by including task T12. If capacity is not exceeded, then PA3 responses as propose to schedule T12 (Figure 7).

New $U3 = [U3 + U12] = [0.8000 + 0.1428] = 0.9428$.

AA deals with the PA1 and accept the proposal of PA1, because new capacity of processor 1 is highest than new capacity of processor 3. AA rejects the proposal of PA3. PA1 informs to AA for reallocation of task T12.

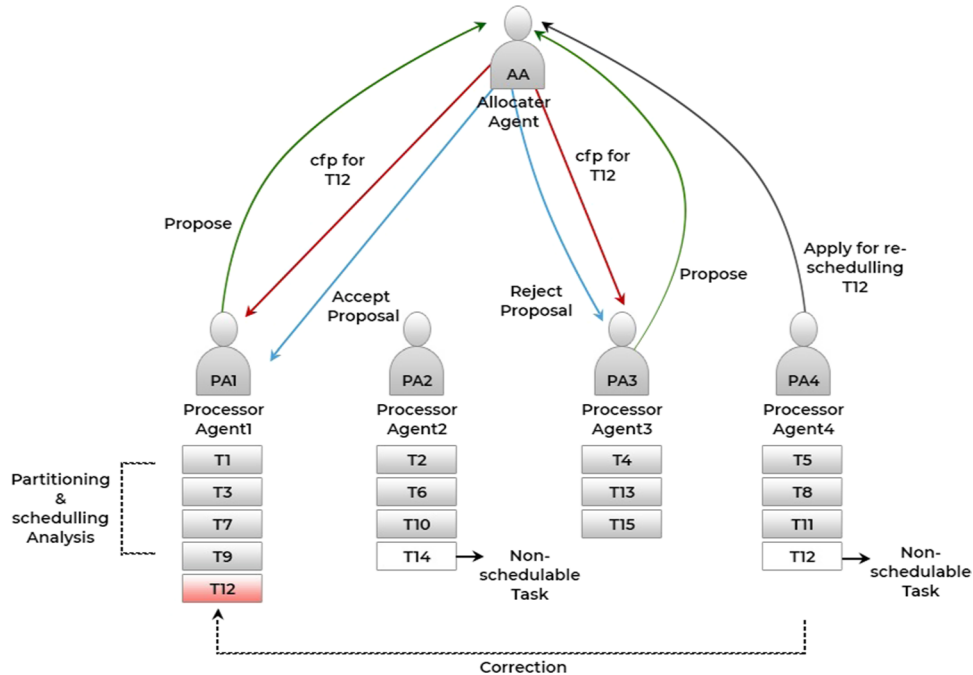


Figure 7: Sequence of CFP for re-scheduling of T12.

Now capacity of 4 processors after rescheduling of T12:

$$U1 = 0.8125 + 0.1428 = 0.9553$$

$$U2 = 1.1111$$

$$U3 = 0.8000$$

$$U4 = 1.1428 - 0.1428 = 1.0000$$

In Figure 8, we assume the contract net protocol with four timing parameters like call-time with specific time 2 sec, response-time with specific time 4 sec, accept/reject-proposal-time with specific time 3 sec, and delivery time with specific time 5 sec. Figure 9 specifies that AA CFP to PA1 and PA3 before or within 2 sec. PA1 and PA3 give the response as accept the proposal at the same time before or within 4 sec. Now at the same time, AA sends a message to PA1 for accept the proposal and also sends a message to PA3 for reject proposal before or within 3 sec. PA1 informs to AA for task reallocation within 5 sec.

FIPA messages

Agents communication in the form of real-time FIPA performatives are given below.

(1) Allocator agent (AA) sends call for proposal (CFP) to PA1 to reallocate the task (T12).

(CFP

:sender AA

:receiver PA3

:content (action(PA3) (Reallocate the task T12))

:call-time(2)

:response-time(4)

:reject-proposal-time(3)

:delivery-time(5)

:language _pa-sl

(2) PA1 is ready to propose for the task (T12) scheduling within response time.

(propose

:sender PA3

:receiver AA

:content(action(AA) (Ready to scheduling T12))

:response-time(4)

:language _pa-sl)

(3) AA approves the proposal of PA1.

(accept-proposal

:sender AA

:receiver PA1

:content “((action (PA1) (Schedule the T12)) (New Capacity of processor U1=0.9553 (Heigh)))”

:accept-proposal-time(3)

:language _pa-sl)

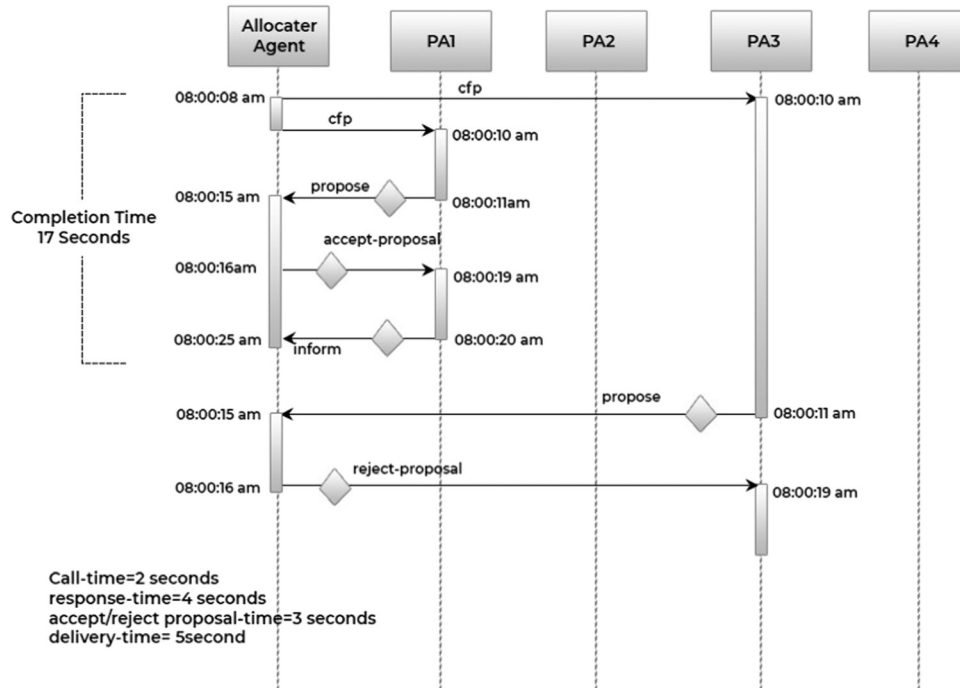


Figure 8: Sequence of proposed time duration of CFP for partition P4.

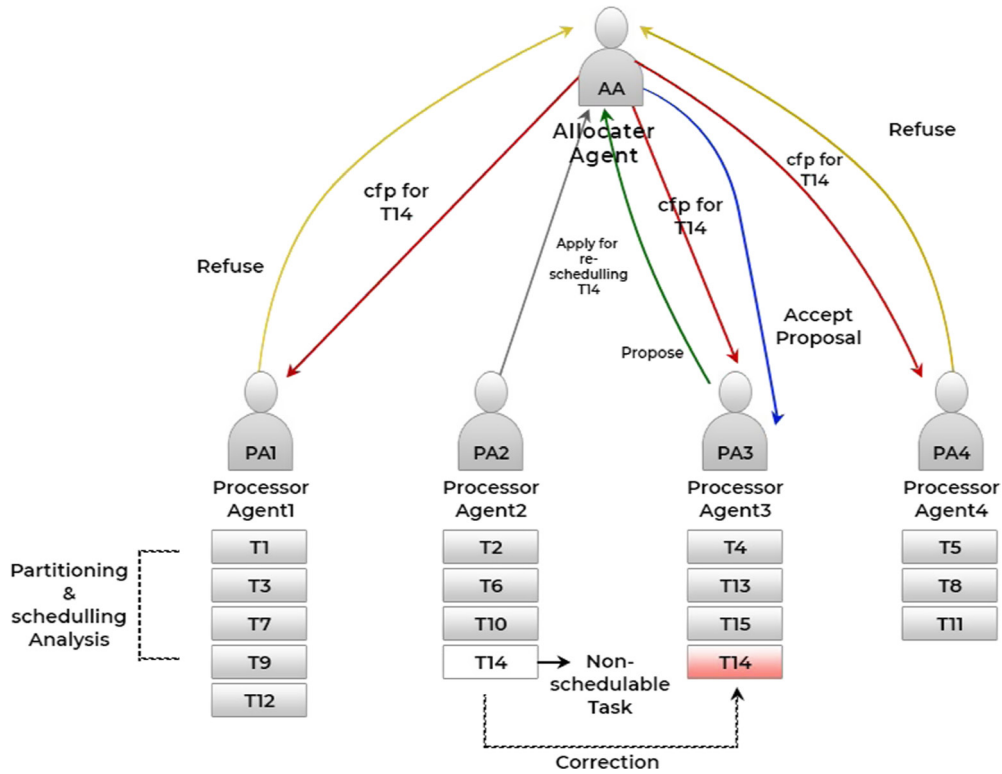


Figure 9: Sequence of CFP for re-scheduling of T14.

(4) Agent allocator deals with the PA1. PA1 informs to AA that task (T12) has been scheduled within delivery time.

```
(inform
:sender PA1
:receiver AA :content "(action (AA) Scheduling
(PA1 schedules T12))"
:delivery-time(5)
:language _pa-sl
```

(5) Allocator agent (AA) sends call for proposal (CFP) to PA3 to reallocate the task (T12).

```
(cfp
:sender AA
:receiver PA3
:content(action(PA3) (Reallocate the task T12))
:call-time(2)
:response-time(4)
:reject-proposal-time(3)
:delivery-time(5)
:language _pa-sl
```

(6) PA3 is ready to propose for the task (T12) scheduling within response time.

```
(propose
:sender PA3
```

```
:receiver AA
:content(action(AA) (Ready to scheduling))
:response-time(4)
:language fipa-sl
```

(7) AA rejects the proposal of PA3.

```
(reject-proposal
:sender AA
:receiver PA3
:content "((action (PA3) (Schedule the T12))(New
Capacity of processor U3=0.9428
(Low)))"
:reject-proposal-time(3)
:language fipa-sl)
```

In Table 1, AA sends a CFP to PA1 and PA3 within 2sec and after receiving CFP, PA1, and PA3 response as propose within 4sec. AA accepts proposal PA1 within 3sec and at the same time AA rejects proposal PA3 within 3sec. PA1 informs to AA for task reallocation within 5 sec. Total time duration of this scenario for CFP completion between many agents will be 17sec. Contract net protocol cannot handle timing aspects with deadlines in RTMAss. In this scenario, agents required real-time performatives during interaction with timing

Table 1. Timing duration of agents in CFP for partition P4.

Sender	Receiver	Actions	Timer	Call-time	Resp-time	Accept/reject-time	Delivery-time	Completion-time
AA	PA1	cfp	08:00:08 AM	2	NA	NA	NA	08:00:10 AM
AA	PA3	cfp	08:00:08 AM	2	NA	NA	NA	08:00:10 AM
PA1	AA	Propose	08:00:11 AM	NA	4	NA	NA	08:00:15 AM
PA3	AA	Propose	08:00:11 AM	NA	4	NA	NA	08:00:15 AM
AA	PA1	Accept-proposal	08:00:16 AM	NA	NA	3	NA	08:00:19 AM
AA	PA3	Reject-proposal	08:00:16 AM	NA	NA	3	NA	08:00:19 AM
PA1	AA	Inform	08:00:20 AM	NA	NA	NA	5	08:00:25 AM

deadlines. AA deals with the PA4 for the reallocation of task T14.

As shown in Figure 9, PA2 calls the AA to reallocate the task T14 and sends it as a cfp to the PA1, PA3, and PA4. T14 ($u_{14}=0.1111$). PA1 calculates the new capacity of processor1 by including task T14. Capacity is exceeded then PA1 does not response as propose to schedule T14.

$$\text{New } U1 = [U1 + U14] = [0.9553 + 0.1111] = 1.0664.$$

PA3 also calculates the new capacity of processor3 by including task T14. If capacity is not exceeded, then PA3 responds as propose to schedule T14.

$$\text{New } U3 = [U3 + U14] = [0.8000 + 0.1111] = 0.9111.$$

PA4 also calculates the new capacity of processor4 by including task T14. Capacity is exceeded then PA4 does not respond as propose to schedule T14.

$$\text{New } U4 = [U4 + U14] = [1.1428 + 0.1111] = 1.2538.$$

Now capacity of 4 processors after rescheduling of T14:

$$U1 = 0.9553$$

$$U2 = 1.1111 - 0.1111 = 1.0000$$

$$U3 = 0.8000 + 0.1111 = 0.9111$$

$$U4 = 1.0000$$

AA deals with the PA3 and accept the proposal of PA3, because new capacity of processor 3 is highest than new capacity of processor 1 and new capacity of processor 4. PA3 informs to AA for reallocation of task T14.

Figure 10 specifies that AA CFP to PA1, PA3, and PA4 before or within 4sec. PA1 and PA4 gave the response as refusing to scheduling of task T14 and at the same time only PA3 gives the response as propose to scheduling of task T14 within the specific time 3sec. AA accepts the proposal of PA3 within specific time 5sec. Now PA3 informs AA to reallocation of task T14 within 5sec. Agents communication in the form of real-time FIPA performatives are specified below.

(1) Allocator agent (AA) sends CFP to PA3 to reallocate the task (T14).

```
(CFP
:sender AA
:receiver PA3
:content(action(PA3) (Reallocate the task T14))
:call-time(4)
:response-time(3)
:reject-proposal-time(5)
:delivery-time(5)
:language FIPA-sl)
```

(2) PA3 is ready to propose for the task (T14) scheduling within response time.

```
(propose
:sender PA3
:receiver AA
:content(action(AA) (Ready to scheduling T14))
:response-time(3)
:language FIPA-sl)
```

(3) AA approves the proposal of PA3.

```
(accept-proposal
:sender AA
:receiver PA3)
```

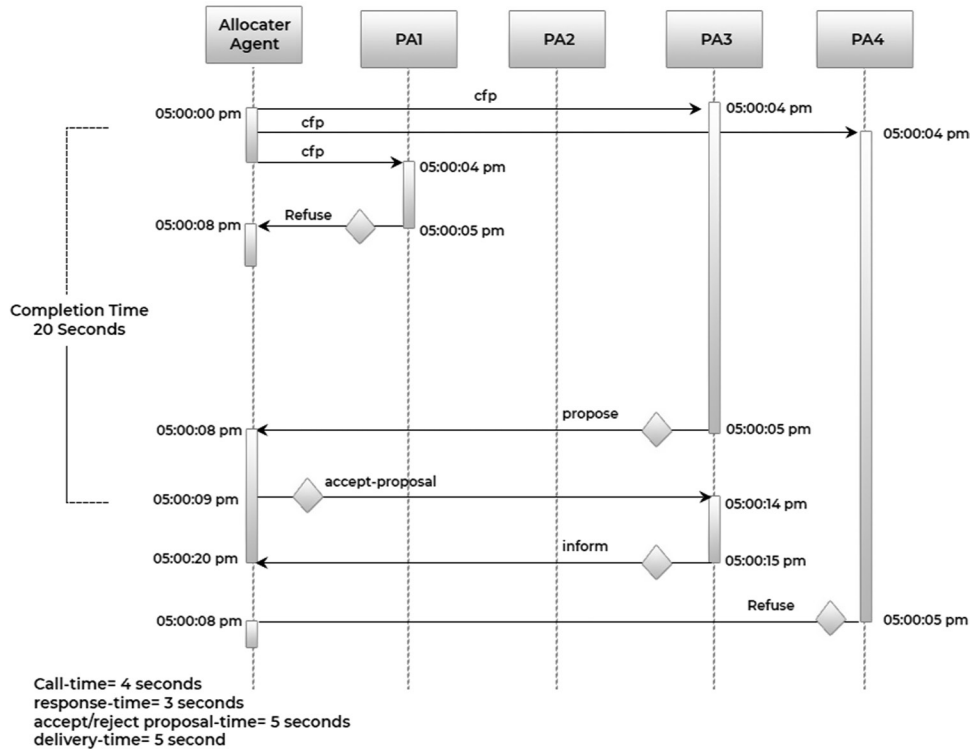


Figure 10: Sequence of proposed time duration of CFP for partition P2.

:content “((action (PA3) (Schedule the T14)) (New Capacity of processor U1 = 0.9111 (Heigh)))”
:accept-proposal-time(5)
:language FIPA-sl)

(4) AA deals with the PA3. PA3 informs to AA that task (T14) has been scheduled within delivery time.

(inform
:sender PA3
:receiver AA
:content “(action (AA) Scheduling (PA3 schedules T12))”
:delivery-time(5)
:language FIPA-sl)

(5) AA sends call for proposal (CFP) to PA1 to reallocate the task (T14).

(CFP
:sender AA
:receiver PA1
:content(action(PA1) (Reallocate the task T14))
:call-time(4)
:response-time(3)

:reject-proposal-time(5)
:delivery-time(5)
:language FIPA-sl)

(6) PA1 is refused to the task (T14) scheduling within response time.

(refuse
:sender PA1
:receiver AA
:content(action(AA) (ready to schedulingT14))
:response-time(3)
:language FIPA-sl)

(7) AA sends CFP to PA4 to reallocate the task (T14).

(CFP
:sender AA
:receiver PA4
:content “((action (PA4) (Schedule the T14)))”
:call-time(4)
:response-time(3)
:reject-proposal-time(5)
:delivery-time(5)
:language _pa-sl)

Table 2. Timing duration of agents in CFP for partition P2.

Sender	Receiver	Actions	Timer	Call-time	Resp-time	Accept/reject-time	Delivery-time	Completion-time
AA	PA1	CFP	05:00:00 PM	4	NA	NA	NA	05:00:04 PM
AA	PA3	CFP	05:00:00 PM	4	NA	NA	NA	05:00:04 PM
AA	PA4	CFP	05:00:00 PM	4	NA	NA		05:00:04 PM
PA1	AA	Refuse	05:00:05 PM	NA	3	NA	NA	05:00:08 PM
PA3	AA	Propose	05:00:05 PM	NA	3	NA	NA	05:00:08 PM
PA4	AA	Refuse	05:00:05 PM	NA	3	NA	NA	05:00:08 PM
AA	PA3	Accept-proposal	05:00:09 PM	NA	NA	5	NA	05:00:14 PM
PA3	AA	Inform	05:00:15 PM	NA	NA	NA	5	05:00:20 PM

(8) PA4 is ready to refuse to the task (T14) scheduling within response time.

```
(refuse
:sender PA4
:receiver AA
:content(action(AA) (no ready to schedulingT14)
:response-time(3)
:language _pa-sl)
```

In Table 2, AA sends a CFP to PA1, PA3 and PA4 within 4 sec. After receiving CFP, PA1 and PA4 respond as refuse and at the same time PA3 respond as propose within 3 sec. AA accepts proposal PA3 within 5 sec and at the same time. PA3 informs to AA for task reallocation within 5 sec. Total time duration of this scenario for CFP completion between many agents will be 20 sec.

Discussion

The main purpose of the proposed real-time FIPA performatives is to enhance the expressivity of standardized performatives by incorporating the ability to handles timing constraints at the messages level. In case study 1, detection of unauthorized boats in marine reserves was time critical. Usage of real-time FIPA performatives made it possible to identify the unauthorized boats within specific time duration (seconds) and relevant authority take the position to capture that unauthorized boats in protected area. Same in scenario 2, fault handling in GPS device in boat4 within minimum time duration (seconds) by asking administrator was elaborated. The second case study of scheduling analysis and correction for non-schedulable tasks in RTMAS was used for correction of non-schedulable partitions P2 and

P4. Correction for non-schedulable partition takes specific time in which partition was re-scheduled.

Conclusion

In this research, we have proposed real-time FIPA performatives for effective functioning of MASs in critical environment. Agent communication is an important characteristic of RTMASs and therefore there is a need of standardized interaction protocols that can be used to handle timing aspects in RTMAS's negotiation. There is no provision to specify deadline at the messages level in existing FIPA performatives and the proposed FIPA performatives compensates it. In our proposed FIPA performatives, timing parameters are introduced through which communication via message passing will help to enhance the overall performance of the system. We demonstrated the effectiveness of the proposed approach using two agent-based real-time case studies, i.e. monitoring boats in marine reserves and scheduling analysis and correction for non-schedulable tasks. The research provides well-defined communicative actions of agents within timing deadlines. In future we will work on extending JADE simulation tool to provide a platform for simulation of these real-time FIPA performatives.

Literature Cited

Bhardwaj, A., Singh, V. K. and Kumar, P. 2014. Multi-agent based train passing in railway system with minimum system delay. 2014 IEEE International Advance Computing Conference (IACC), 855–859, available at: <https://doi.org/10.1109/IAdCC.2014.6779434>.

- Bhouri, N., Balbo, F. and Pinson, S. 2012. An agent-based computational approach for urban traffic regulation. *Progress in Artificial Intelligence* 1(2): 139–147, available at: <https://doi.org/10.1007/s13748-012-0011-0>.
- Botti, V., Carrascosa, C., Julián, V. and Soler, J. 1999. Modelling agents in hard real-time environments. European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Springer, 63–76, available at: https://doi.org/10.1007/3-540-48437-X_6.
- Cao, M. 2012. Multi-strategy selection supported automated negotiation system based on BDI agent. 2012 45th Hawaii International Conference on System Sciences, IEEE, 638–647, available at: <https://doi.org/10.1109/HICSS.2012.442>.
- Cintuglu, M. H., Martin, H. and Mohammed, O. A. 2015. An intelligent multi agent framework for active distribution networks based on IEC 61850 and FIPA standards. 2015 18th International Conference on Intelligent System Application to Power Systems (ISAP), IEEE, 1–6, available at: <https://doi.org/10.1109/ISAP.2015.7325527>.
- Elshaafi, H., Vinyals, M., Grimaldi, I. and Davy, S. 2018. Secure automated home energy management in multi-agent smart grid architecture. *Technology and Economics of Smart Grids and Sustainable Energy* 3(1): 1–13, available at: <https://doi.org/10.1007/s40866-018-0042-0>.
- Filgueiras, T. P., Lung, L. C. and de Oliveira Rech, L. 2012. Providing real-time scheduling for mobile agents in the jade platform. 2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, IEEE, 8–15, available at: <https://doi.org/10.1109/ISORC.2012.10>.
- Ghorbani, J., Choudhry, M. A. and Feliachi, A. 2012. Real-time multi agent system modeling for fault detection in power distribution systems. 2012 North American Power Symposium (NAPS), IEEE, 1–6, available at: <https://doi.org/10.1109/NAPS.2012.6336398>.
- Ghosn, S. B., Ranganathan, P., Salem, S., Tang, J., Loegering, D. and Nygard, K. E. 2010. Agent oriented designs for a self-healing smart grid. 2010 First IEEE International Conference on Smart Grid Communications, IEEE, 461–466, available at: <https://doi.org/10.1109/SMARTGRID.2010.5622085>.
- Hosny Abbas, M. A. and Shaheen, S. 2015. On the adoption of multi-agent systems for the development of industrial control networks: a case study. Proceeding of The Eleventh International Conference on Autonomic and Autonomous Systems, ICAS, pp. 24–29, available at: <https://arxiv.org/abs/1506.05235>.
- Jemal, H., Kechaou, Z., Ayed, M. B. and Alimi, A. M. 2015. A multi agent system for hospital organization. *International Journal of Machine Learning and Computing* 5(1): 51–56, available at: <https://doi.org/10.7763/IJMLC.2015.V5.482>.
- Julian, V. and Botti, V. 2004. Developing real-time multi-agent systems. *Integrated Computer-Aided Engineering* 11(2): 135–149, available at: <https://doi.org/10.3233/ICA-2004-11204>.
- Julian, V., Carrascosa, C., Rebollo, M., Soler, J. and Botti, V. 2002. SIMBA: an approach for real-time multi-agent systems. Catalanian Conference on Artificial Intelligence, Springer, 282–293, available at: https://doi.org/10.1007/3-540-36079-4_25.
- Kexing, L. 2011. A survey of agent based automated negotiation. 2011 International Conference on Network Computing and Information Security, IEEE, Vol. 2, 24–27, available at: <https://doi.org/10.1109/NCIS.2011.103>.
- Kristensen, T. and Smith, K. 2015. Intelligent traffic simulation by a multi-agent system. 2015 Third World Conference on Complex Systems (WCCS), IEEE, 1–7, available at: <https://doi.org/10.1109/ICoCS.2015.7483321>.
- Li, W., Logenthiran, T., Woo, W. L., Phan, V. -T. and Srinivasan, D. 2016. Implementation of demand side management of a smart home using multi-agent system. 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2028–2035, available at: <https://doi.org/10.1109/CEC.2016.7744037>.
- Linnenberg, T., Wior, I., Schreiber, S. and Fay, A. 2011. A market-based multi-agent-system for decentralized power and grid control. ETFA 2011, IEEE, 1–8, available at: <https://doi.org/10.1109/ETFA.2011.6059126>.
- Mahfoudhi, A., Karamti, W. and Zaguia, A. 2018. Scheduling analysis and correction for multiprocessor real-time systems based on multi-agent systems. *International Journal of Applied Engineering Research* 13(5): 2368–2374.
- Qasim, A. and Kazmi, S. A. R. 2016. Mape-k interfaces for formal modeling of real-time self-adaptive multi-agent systems. *IEEE Access* 4: 4946–4958, available at: <https://doi.org/10.1109/ACCESS.2016.2592381>.
- Qasim, A., Kazmi, S. A. R. and Fakhir, I. 2015. Formal specification and verification of real-time multi-agent systems using timed-arc petri nets. *Advances in Electrical and Computer Engineering* 15(3): 73–78, available at: <https://doi.org/10.4316/AECE.2015.03010>.
- Qasim, A., Kanwal, S., Khalid, A., Kazmi, S. A. R. and Hassan, J. 2019. Timed-arc petri-nets based agent communication for real-time multi-agent systems. *International Journal of Advanced Computer Science and Applications* 10(9): 529–537, available at: <https://doi.org/10.14569/IJACSA.2019.0100970>.
- Qasim, A., Aziz, Z., Kazmi, S. A. R., Khalid, A., Fakhir, I. and Hassan, J. 2020. Intelligent agent for formal modelling of temporal multi-agent systems. *International Journal on Smart Sensing and Intelligent Systems* 13(1): 1–13, available at: <https://doi.org/10.21307/ijssis-2020-003>.
- Reid, M. and Shakshuki, E. M. 2017. Implementing a multi-agent system for recording and transmitting biometric information of elderly citizens. *Procedia Computer Science* 113: 334–343, available at: <https://doi.org/10.1016/j.procs.2017.08.342>.
- Shpilevoy, V., Shishov, A., Skobelev, P., Kolbova, E., Kazanskaia, D., Shepilov, Y. and Tsarev, A. 2013. Multi-agent system “smart factory” for real-time workshop

management in aircraft jet engines production. Proceedings of the 11th IFAC Workshop on Intelligent Manufacturing Systems (IMS'13), 204–209, available at: <https://doi.org/10.3182/20130522-3-BR-4036.00025>.

Stankovic, J. A. 1988. Misconceptions about real-time computing: a serious problem for next generation systems. *Computer* 21(10): 10–19, available at: <https://doi.org/10.1109/2.7053>.